

Application Number 10/617,455
Responsive to Office Action mailed August 11, 2005

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently amended): A method comprising:
computing data that defines a rectangular area of pixels that bounds a triangular area of the pixels;
selecting a line of pixels within the rectangular area of pixels;
sequentially evaluating coordinates associated with ~~of~~ the pixels of the line of pixels~~the~~
rectangular area to determine which ~~whether~~ the pixels fall within the triangle area;
ceasing evaluation of the coordinates associated with the pixels of the line of pixels upon
determining that at least one pixel of the line falls within the triangle area and a current pixel no
longer falls within the triangle area; and
updating pixel data for the pixels that fall within the triangle area to render the triangular area.
2. (Original): The method of claim 1, wherein evaluating coordinates comprises evaluating the coordinates of the pixels in accordance with a set of linear equations for computing edges of the triangular area.
3. (Original): The method of claim 2, wherein evaluating the coordinates of the pixels comprises:
computing a coefficient matrix M_C for computing linear coefficients for the set of linear equations; and
applying the coefficient matrix M_C to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area.
4. (Original): The method of claim 3, wherein applying the coefficient matrix M_C comprises applying the coefficient matrix M_C to a current one of the pixels (X_C , Y_C) within the rectangular area to determine whether:

Application Number 10/617,455
 Responsive to Office Action mailed August 11, 2005

$$M_c \begin{bmatrix} X_c \\ Y_c \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where}$$

the coefficient matrix M_c equals:

$$M_c = \begin{bmatrix} y_1 - y_2 & x_2 - x_1 & x_1 y_2 - x_2 y_1 \\ y_2 - y_0 & x_0 - x_2 & x_2 y_0 - x_0 y_2 \\ y_0 - y_1 & x_1 - x_0 & x_0 y_1 - x_1 y_0 \end{bmatrix} \text{ and}$$

vertices $v_0(x_0, y_0)$, $v_1(x_1, y_1)$, and $v_2(x_2, y_2)$ are vertices of the triangular area.

5. (Original): The method of claim 1, wherein updating pixel data comprises computing pixel data for the pixels of the rectangular area that fall within the triangle area in accordance with a set of linear equations that describe one or more attributes associated with the triangular area.

6. (Original): The method of claim 5, wherein the attribute values comprise at least one of color values and texture values.

7. (Original): The method of claim 5, wherein updating pixel data comprises:
 computing a coefficient matrix M^1 for computing linear coefficients of the set of linear equations; and
 applying the linear coefficients to each of the pixels that falls within the triangular area to compute an attribute value for each of the pixels.

8. (Original): The method of claim 7, wherein applying the coefficient matrix M^1 comprises applying the coefficient matrix M^1 to compute the linear coefficients A, B, and C, for an attribute associated with vertices $v_0(x_0, y_0)$, $v_1(x_1, y_1)$, and $v_2(x_2, y_2)$ of the triangle as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = M^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix},$$

Application Number 10/617,455
 Responsive to Office Action mailed August 11, 2005

where the coefficient matrix M^{-1} equals:

$$M = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \text{ and}$$

an attribute value for each pixel (X_c, Y_c) is computed as

$$v = AX_c + BY_c + C.$$

9. (Original): The method of claim 1, further comprising:
 determining whether each of the pixels within the rectangular area is visible; and
 selectively rendering each of the pixels based on the determination.

10. (Original): The method of claim 9, wherein determining whether each of the pixels is visible comprises comparing a z-value, z_c , of the current pixel with a corresponding z-value, z_b , of a z-buffer to determine if $z_c < z_b$.

11. (Original): The method of claim 1, wherein computing data comprises computing a first coordinate and a second coordinate that represent opposite corners of the rectangular area.

12. (Original): The method of claim 1, wherein
 updating pixel data comprises processing the pixel data within a cache memory having a block size; and

computing data to define a rectangular area comprises computing the data to define the rectangular area as a function of the block size of the cache.

13. (Currently amended): An apparatus comprising a rendering engine that defines a rectangular area of pixels that bounds a triangular area of the pixels, selects a line of pixels within the rectangular area of pixels, sequentially and evaluates coordinates associated with the pixels of the line of pixels to determine whether the pixels fall within the triangle area, and ceases evaluation of the coordinates associated with the pixels of the line of pixels upon determining that

Application Number 10/617,455
 Responsive to Office Action mailed August 11, 2005

at least one pixel of the line falls within the triangle area and a current pixel no longer falls within the triangle area ~~rectangular area to selectively render the pixels that fall within the triangular area.~~

14. (Original): The apparatus of claim 13, wherein the rendering engine evaluates the coordinates of the pixels in accordance with a set of linear equations that describe edges of the triangular area.

15. (Original): The apparatus of claim 14, wherein the rendering engine computes a coefficient matrix M_C for computing linear coefficients for the set of linear equations, and applies the coefficient matrix M_C to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area.

16. (Original): The apparatus of claim 15, wherein the rendering engine applies the coefficient matrix M_C to a current one of the pixels (X_C , Y_C) within the rectangular area to determine whether:

$$M_C \begin{bmatrix} X_C \\ Y_C \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ where}$$

the coefficient matrix M_C equals:

$$M_C = \begin{bmatrix} y_1 - y_2 & x_2 - x_1 & x_1 y_2 - x_2 y_1 \\ y_2 - y_0 & x_0 - x_2 & x_2 y_0 - x_0 y_2 \\ y_0 - y_1 & x_1 - x_0 & x_0 y_1 - x_1 y_0 \end{bmatrix} \text{ and}$$

vertices $v_0(x_0, y_0)$, $v_1(x_1, y_1)$, and $v_2(x_2, y_2)$ are vertices of the triangular area.

17. (Original): The apparatus of claim 13, wherein the rendering engine selectively renders the pixels that fall within the triangular area by computing updated pixel data for those pixels in accordance with a set of linear equations that describe one or more attributes associated with the triangular area.

Application Number 10/617,455

Responsive to Office Action mailed August 11, 2005

18. (Original): The apparatus of claim 17, wherein the attribute values comprise at least one of color values and texture values.

19. (Original): The apparatus of claim 17, wherein the rendering engine computes a coefficient matrix M^{-1} for computing linear coefficients A, B, C of the set of linear equations, and applies the coefficients A, B, C to each pixel that falls within the triangular area to compute an attribute value for the respective pixel.

20. (Original): The apparatus of claim 19, wherein the rendering engine applies the coefficient matrix M^{-1} to compute the linear coefficients A, B, C, for an attribute associated with vertices $v_0(x_0, y_0)$, $v_1(x_1, y_1)$, and $v_2(x_2, y_2)$ of the triangle as:

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = M^{-1} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix},$$

where the coefficient matrix M^{-1} equals:

$$M = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}, \text{ and}$$

an attribute value for each pixel (X_c, Y_c) is computed as

$$v = AX_c + BY_c + C.$$

21. (Original): The apparatus of claim 13, further comprising a z-buffer storing a set of z-values associated with the pixels, and wherein the rendering engine compares a z-value, z_c , of the current pixel with a corresponding z-value, z_b , of a z-buffer to determine whether each pixel within the rectangular area is visible and selectively renders each pixel of the rectangular area that is visible and that falls within the triangle area.

22. (Original): The apparatus of claim 13, further comprising a control unit that issues a command to the rendering engine that specifies vertices of the triangular area.

23. (Original): The apparatus of claim 13, wherein the rendering engine comprises:

Application Number 10/617,455
Responsive to Office Action mailed August 11, 2005

a vertex buffer for buffering the vertices of the triangular area to be rendered;
a bounding box generator that processes the vertices to compute bounding data that define the dimensions of the rectangular area; and
a rasterizer that processes the bounding data and evaluates coordinates associated with the pixel values of the rectangular area to selectively render the pixels that fall within the triangular area.

24. (Original): The apparatus of claim 23, further comprising:

an edge coefficient generator that receives the vertices buffered by the vertex buffer and processes the vertices to compute linear coefficients for a set of linear equations that describe edges of the triangular area, and

an attribute coefficient generator that processes the vertices to compute linear coefficients for a set of linear equations that describe one or more attributes associated with the triangular area, wherein

the rasterizer processes the bounding data and the coefficients in accordance with the sets of linear equations to render the pixels that fall within the triangular area.

25. (Original): The apparatus of claim 13, wherein the apparatus comprises a wireless communication device.

26. (Original): The apparatus of claim 13, wherein the apparatus comprises an integrated circuit.

27. (Original): The apparatus of claim 13, further comprising a cache memory to store at least a portion of the pixels, wherein the cache memory has a block size, and the rendering engine defines the rectangular area as a function of the block size of the cache.

28. (Currently amended): A mobile communication device comprising:

a display;
a processor to generate video output data for presentation by the display as a graphical environment; and

Application Number 10/617,455
Responsive to Office Action mailed August 11, 2005

a rendering engine that applies a direct evaluation algorithm to render a triangle for the graphical environment, wherein the direct evaluation algorithm applies a coefficient matrix defining a set of linear coefficients for a set of linear equations that describe edges of the triangle to pixels~~linear equations~~ to render the triangle without interpolating between edges of the triangle.

29. (Original): The mobile communication device of claim 28, wherein the processor issues a command to the rendering engine that defines vertices for the triangle.

30. (New): The method of claim 1, further comprising:
selecting a subsequent line of pixels within the rectangular area of pixels;
sequentially evaluating coordinates of pixels of the subsequent line of pixels to determine whether the pixels fall within the triangle area;
ceasing evaluation of the coordinates of the pixels of the subsequent line of pixels upon determining that at least one pixel of the subsequent line falls within the triangle area and a current pixel no longer falls within the triangle area; and
updating pixel data for the pixels that fall within the triangle area to render the triangular area.

31. (New): A method comprising:
computing data that defines a rectangular area of pixels that bounds a triangular area of the pixels;
computing a coefficient matrix for computing a set of linear coefficients of a set of linear equations that describe edges of the triangular area;
applying the coefficient matrix to each of the pixels within the rectangular area to determine whether each of the pixels falls within the triangular area; and
updating pixel data for the pixels that fall within the triangle area to render the triangular area.